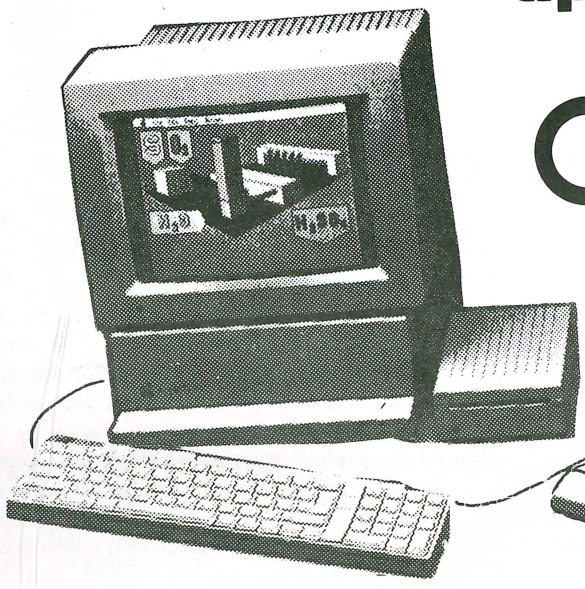
THE SEED

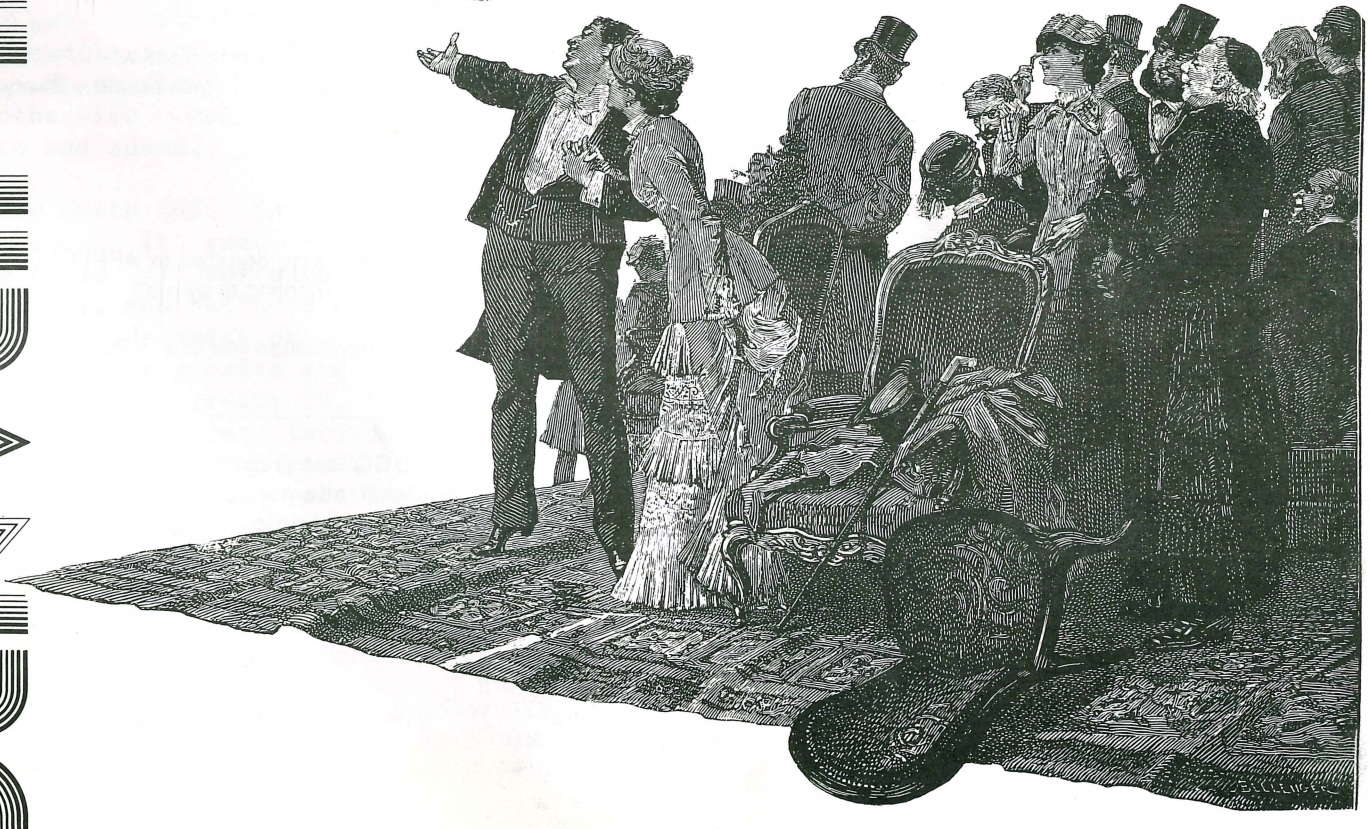
October 1986
Vol. 8 No. 10

NEWSLETTER FOR APPLE PEOPLE

DENVER APPLE PI

apple's Granny Smith





denver apple pi Seed

P.O. BOX 17467
DENVER, COLORADO 80217

President .....Roy Dean
Vice President & Property .....Trent Hein
Secretary & Editor .....Randall Ansley
Treasurer .....Bob Hein
Board Member & Software Librarian .....Ed Thompson
Board Member & Store Rep .....Tom Angelos
Board Member .....Bill Cronister
Sales Officer .....Chris Saxon
BBS Officer .....Robert Racansky

MEMBERSHIP INFORMATION

APPLE PI is an Apple Computer Users Group - not affiliated with Apple Computer, Inc. - that provides many member benefits, some of which are:

MONTHLY MEETINGS - On the first Thursday of each month at 7:00 p.m. APPLE PI holds its general meeting at the Green Center, Colorado School of Mines Campus, Golden, Colorado. This meeting is a combination of an informational meeting, new product announcements, and a short business meeting. The members are given an opportunity to buy computer supplies a little above cost. Software products are given away as door prizes.

On the Sunday following the general meeting there is an informal meet which is usually held at the Clear Creek Office Park, 4251 Kipling, in Wheat Ridge. This meeting is from 10:00 a.m. until 3:00 p.m. and provides a more informal environment to seek or give advice to fellow members and get copies of programs from the APPLE PI Library. Both of these meetings are open to the general public but only members have access to the club library and products that are sold.

PROGRAM LIBRARY - APPLE PI currently has a library of 9,000 programs. All current disks are in 16 sector (3.3, Pascal, CP/M) format and are available to any member. APPLE PI Members can obtain copies from the library:

1) By coming to the monthly informal meeting (swap meet);
2) By written request to the club librarian (at any meeting or by mail);
3) By electronic mail on APPLE PI's bulletin board system:

DISK OF THE MONTH - Each month a collection of some of the best programs donated to APPLE PI during the preceding month are made available. The DOM can be purchased at any group meeting or by mail.

DUES - Current membership dues are **$18.00** per year, with a one-time application fee of **$10.00.** Checks should be made payable to DENVER APPLE PI.

The words APPLE, APPLE COMPUTER, and the APPLE LOGO are registered trademarks of the Apple Computer Company Inc. Representation of these trademarks in the Seed does not constitute any expression of agreement or disagreement by Apple Computer Inc. with any articles or other printed material found in the Seed. The opinions, policies and comments contained in the Seed are those of the Apple Pi users group, officers and members.

NOTICE

Articles, stories, and other items in the Seed may be copied or used by other clubs or groups as long as credit is given to Denver Apple Pi and the author of the article. On reprints please include the date or volume number from the Seed.

ADVERTISING RATES

1/4 Page	$ 25.00
1/2 Page	$ 40.00
Full Page	$ 70.00

//GS

First came Apple, the company, and with it their first product, the Apple computer. Less than 200 Apples were ever made. Not much more than a circuit board, the Apple was soon replaced by the Apple][. After many improvements in the Apple][, an all new business machine was introduced called the Apple ///. When the Apple /// failed to produce the acceptance in the business community that Apple had hoped for, the company's attention was turned to the Lisa and eventually the Macintosh for the business computer. The Apple][became the][+, and about three and a half years ago it became the //e, the biggest selling computer in Apple's arsenal. A compact version, the //c, followed, and the Mac has evolved into the Mac+.

Apple users today are split into two camps, the 6502'ers and the 68000'ers. Mac folks like the graphics and ease of using the Mac while the]['ers like the fact that the 6502 machines are easy to program and have such a good selection of inexpensive software available. Well, now, the split is over...We welcome the new "Granny Smith" Apple, otherwise known as the Apple //gs (graphics and sound).

The Granny Smith (GS) is the computer for all comers. It runs almost all the existing Apple // software but has the speed, memory, and built in goodies that we have all been dreaming of. Granny comes with slots to put goodies into, but most of the goodies are already in the machine. Some of these goodies include 2 serial ports, a clock, a drive port, networking, 80 columns, and 256k of RAM on the motherboard. Actually there is 320k on board but 64k is used by the incredible sound system. The Apple //e has been crunched down to one chip, YES ONE CHIP!!! and the machine now has the ability to run at 2.5 times the speed of a //e and has a 16 bit addressing ability because of the use of the W65SC816 microprocessor from Western Design Center.

I will not go into any more details of the new Granny Smith here. Apple Pi has a real treat for you. Our guest speakers for the October meeting will be Apple Computer Inc., and you will get to see the Granny Smith live and in person. Mark your calendar for October 2nd. You will love this computer.

New Association

Apple Programmers and Developers Association Formed

Apple Computer, Inc. and A.P.P.L.E. Co-op have announced the formation of the Apple Programmers and Developers Association (APDA). This new association is intended to provide a central source of professional software and hardware development tools, technical notes, and technical manuals for those developing software and peripherals for any of Apple's computers.

The APDA membership will include a quarterly catalog and newsletter, regular update bulletins, and ordering priviledges. The membership will be $20.00 per year.

APDA will inventory, catalog, and distribute development tools, system software, technical notes, utilities and manuals from Apple, languages and development tools from a variety of third-party companies, and technical books, including the entire Apple Technical Library from Addison-Wesley.

APDA is not for everyone. It is not a national user's group. It is an organization for the technically oriented Apple user. It supplements local user's groups such as Denver Apple Pi. Only time will tell how effective it will be, but its intent is to bring back that original close tie that Apple had with its users.

If you wish to become an APDA member, call (206) 251-6548 or write to APDA at 290 SW 43th Street, Renton, Washington, 98055.

PRESIDET'S REPORT

Fall brings many thoughts to many different people. Some are thinking of the World Series or the start of football. Other's look forward to the aspen's rich colors. Still others think of skiing or dread the on coming winter. Here at Apple Pi, we look forward to the fall as that time of year when people are back to school or work and the attendance at meetings increases. It always seems as if the club builds during the fall. We are looking forward to seeing all of you who have had a summer vacation or are a new member. Oh yes, those of you who are always at all the meetings, it's always great to see your bright faces in the crowd.

We have several activities in process for this month and the rest of the year. First, this month we are having a garage sale. This is a great time for you to get rid of any of that "out dated" equipment you have around the house or office. Bring in any computer hardware, software (originals only), trinkets, toys, or other computer paraphernalia. The garage sale will be held at the core meet on October 5 at the Clear Creek Office Park, 4251 Kipling. Remember, "One man's junk is another man's treasure." Hope to see you there.

Also, as those of you that have attended the last meetings know, the Lakewood Library has provided room for Apple Pi to have its hardcopy library available. The material will be on a reference only status, but there are plenty of copy machines available for your convenience. I am asking members of Apple Pi to help make this a super hardcopy library. You can help by donating any old magazines, newsletters, pamphlets, text, manuals, etc. to the club for use in the hardcopy library. If you have any of that material laying around that your not using any more, please bring it to either the general meeting or the core meeting and I will collect it and take it to the library. I hope for this to be an on going process. The club will continue to add its hardcopy to the library on a monthly basis. You can help by continuing to donate (remember, it's tax deductible) to the club's library. With your support, we can have a superior reference library.

On a final note, just a reminder that Apple Computer Inc. is scheduled for the October meeting to show off the //gs. If you have not gotten a look at Apple's new computer, then the October meeting will be a great opportunity for you. I hope to see you all at the meetings.

Sincerely,

Roy B. Dean
President

LIBRARIAN'S REPORT

Since October 1 is the third aniversary of Apple's introduction of ProDOS, this month's DOM will be, guess what, ProDOS! We have previously offered two ProDOS DOM's (April and May 1986) and have collected another disk of really excellent programs. Without getting into the debate over its advantages and disadvantages, ProDOS is rapidly becoming the operating system of choice for Apple users and programmers.

The disk label below lists the contents of the October DOM. The feature programs are Diskworks, The Executive, and Propaker. Diskworks is a ProDOS block editor offered as shareware by Living Legends. It requires 80 columns and self configures to work with volume sizes from 280 blocks to 30 megabytes. Its features include search, dissassemble, display as hex or ascii, dump to printer, hex decimal converter, and hex or ascii data entry.

Denver Apple Pi's Disk of the Month
#075 OCT 1986 DOM

STARTUP	MENU	COPYALL
DEVICES	DISKWORKS	DISKWORKS.DOC
FILE.MANAGER	COPY.OBJ0	TYPE.OBJ0
HYPER.FORMAT	PROPAK.STARTUP	PROPAK.SYSTEM
PROPACKER.DOCS	SORT.CATALOG	THE.EXECUTIVE
TIME	CMD.DOCUMENT	COPY.CMD
COPY.CMDS	EXEC.MAKER	FILTER.CMD
FORMAT.CMD	MENU.COPY	MORE.CMD
MOUNT.CMD	TYPE.CMD	UNFORMAT.CMD

The Executive is a communication utility which converts Applesoft, Binary, and even Appleworks files to tokenized upload text files. Propaker is a communication utility which packs and unpacks both ProDOS and DOS 3.3 floppy disks for easy upload/download.

The other programs include Copyall which is a quick full disk copy program, Hyper.Format which will format/verify a ProDOS disk from Basic.system without the system utilities, Devices which will list all active devices by unit number, Time which will set the system date and time without a clock, and Sort.Catalog which will correctly sort the root and all subdirectories of a ProDOS volume. Also included is File Manager which is a utility program to easily copy, type, lock, unlock, delete, rename, and run programs. It also displays the current volume name, the drive in use, and the block usage statistics.

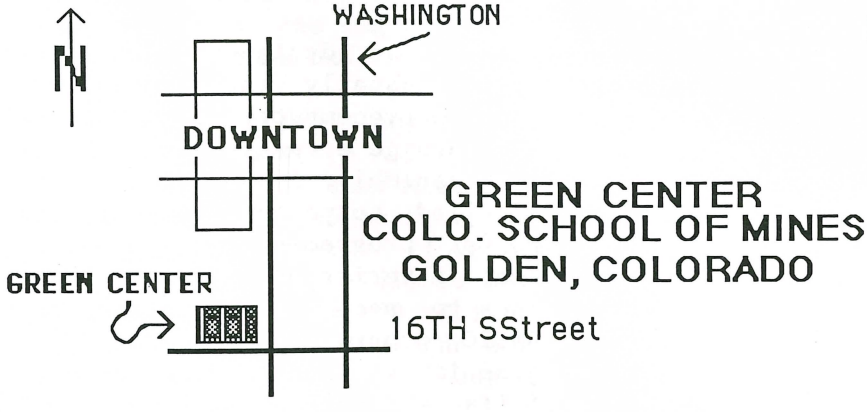
The final group of programs are from Steven Humpage and are primarily add in system commands for ProDOS. These add in commands include an improved Type, Copy, More, Add, Part, Format, Unformat, Filter, and Mount. They perform functions such as concatenating files, splitting files, backing up disks, adding or removing carriage returns from files, and others. Also included are Menu.Copy which is a menu driven utility which displays a directory and permits marking of files to be copied. A companion program, Exec.Maker, will build an exec file to automatically transfer files from one disk to another. Both are excellent for loading a Ramcard.

The back of this month's DOM contains HGR Filer 1.3 by Ken Tanaka, a member of Apple Pi. HGR Filer is a ProDOS utility for managing a library of HIRES pictures and contains a Scrunch routine and hires dump (for Epson). This utility was chosen by the International Apple Core for their DOM number 62 so congratulations to Ken and thanks for the donation.

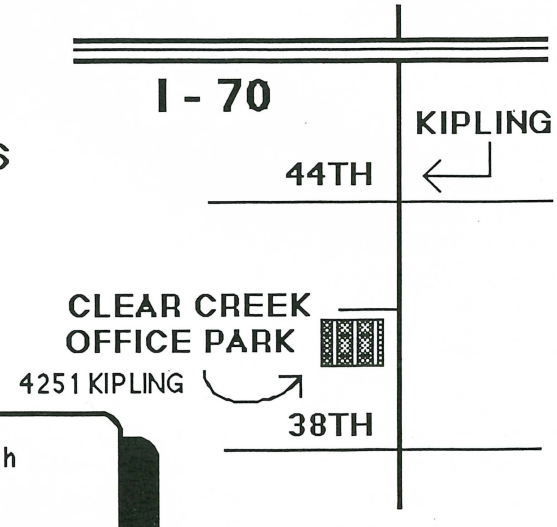
Copies made in September were up considerably from the slow summer at 101 and about 36 DOM's were sold. We also sold out the August DOM and made another dozen. As a special for the October general meeting, we expect to have a collection of 5 double sided print shop graphics type disks complete with pictorial reference guides for $2.50 a disk or the whole set for $10.00.

GOTO 9

GENERAL MEETING



CORE MEET



GENERAL MEETING 7:00 PM first Thursday of each month

CORE MEET 10:00 to 3:00 Sunday after General Meeting

CONFESSIONS OF A BASIC LOVER

I have a certain sympathy for the underdog, that's an American trait. Underdogs get kicked around for things that aren't their fault, and their chances of accomplishing anything are rated pretty low.

Everyone kicks BASIC around, and so my interested is naturaly heightened. Like all underdogs, BASIC has a story to tell, and, sure enough, it's pretty fascinating. For example, did you know that at one time you could buy a computer with 16K of RAM. Now what can you do with sixteen K! Certainly you could't load in a PASCAL compiler, and I bet you couldn't even load in a compiled PASCAL program that would do as much as print your name.

When I bought my first Apple in 1979, it had 32K, a three digit serial number on the disk drive, and a ROM card with Applesoft (I switched the ROMS so that Applesoft was on the motherboard). I learned to program in Applesoft and later in assembly language. Sure, I know the difference between the data types Boolean, Char, Integer and Real, and, in an understated way, so does Applesoft. Try to assign an interger or real number to a string variable and you get a TYPE MISMATCH error. In Applesoft, any numeric variable can be a boolean. Statements like 'IF X THEN 6502' will work just fine, thank you. Any value other than zero in X will cause a jump to line number 6502. It even makes semantic sense to say if X exists (has a value) then do something. Alternately, you can sya if X does not exist do something e.g. 'IF NOT X THEN 6502'.

Of course programming discipline and intellectual rigor are important in a humorless sort of way, and I agree that certain constructs such as DO WHILE and DO UNTIL are missing in BASIC, never mind data structures other than arrays, but how much can you expect from a 10k interpreter? Given the amount of memory it takes these days to do anything 'right', it seems a miracle that microcomputer interpreted BASICS exist at all.

The microcomputer BASICs that we're used to are adapted to a harsher environments than the spoiled rich kid languages that we're supposed to aspire to using. The lush operating systems and fancy editors that PASCAL requires don't come free. In 64K systems a PASCAL compiler may have a disk for the editor and source files, one for the compiler, a third for the object code, and still you have to pray that the 'libraries' that have to be included to get the program to actually work aren't all too large. Woe be unto they that miss a semi-colon!

It should be clear by now that I'm an Applesoft junkie, so much so that I'm even willing to spend the time to see how it actually gets all that stuff done in 10K of code. Taking the time is worth the effort, for possibly never, in the history of computing, have so few resources done so much for so many users. That should be accomplishment enough for hard-nosed educators, even when pretending to be mathematicians, to grudingly admire.

Although BASIC was designed as a compiled language, it's usually interpreted in its better known microcomputer form. This means that the language actually consists of a series of assembly language subroutines that are called to perform general tasks. Compilers (or programmers) on the other hand, create a series of assembly language subroutines to accomplish specific tasks. When a BASIC program is run, BASIC scans the program lines one at a time and decides which of its subroutines to execute. At the beginning of each program line there is overhead to be attended to, such as remembering which linenumber is currently executing. In executing a GOSUB or (horrors) GOTO statement, the interpreter has to go to the beginning of the program looking for the line number to branch to. All this slows things down considerably.

Compiler source code is usually a text file, and since each letter in the file is a character, source files get large rather quickly. Applesoft source files are considerably smaller than their textfile equivalents, a fact that becomes apparent if you have ever created an Applesoft EXEC file. This is because Applesoft is tokenized; each command's reserved word is represented by a single byte code that Applesoft recognizes. When you list a program, each of the byte sized tokens gets expanded to the character string that you type into a program to enter the command, a fact that we will use later to create our own commands.

Applesoft tokens have their high bit set to distinguish them from other characters found in programs that might represent numbers or variable names. For example, the number that corresponds to a byte with only the high bit set, (the smallest possible byte representation of a keyword) is hex 80 or decimal 128. Hex 80 is translated to the character string END by the interpreter. In order to understand Applesoft, we have to be able to look at it, and that we can only do using the monitor. Typing CALL -151 at the keyboard gets us into the monitor, and typing control-C gets us back to Applesoft.

Applesoft programs normally start at address $800, and, to clear the slate, we should first fill some memory starting at $800 with zeros so that anything we do later will stand out. If we get into the monitor (by CALLing -151) we can put a zero at $800 by typing 800:00 (RETurn) at the * prompt. Let's next copy the zero into the memory range $800 to $1002 by typing 801<800.1000M. We can verify that all is clear by typing 800.820 in the monitor to list the first 33 bytes of Applesoft program memory. The CRT should display:

```
0800- 00 00 00 00 00 00 00 00
0808- 00 00 00 00 00 00 00 00
0810- 00 00 00 00 00 00 00 00
0818- 00 00 00 00 00 00 00 00
0820- 00
```

That done we can get back to Applesoft by typing Control-C (RET), and the Applesoft prompt] will reappear. Now we can write a simple program and see what's in memory. The simplest would be a program with a single statement, say, END. Typing it in at the Applesoft (]) prompt:

```
]10 END
```

Next we'll look at it by getting into the monitor, typing:

]CALL -151 (Each typed entry is followed by (RETurn) of course)

The monitor prompt (*) should be displayed and we can look at our program in memory by typing:

```
*800.820
```

```
0800- 00 07 08 0A 00 80 00 00
0808- 00 00 00 00 00 00 00 00
0810- 00 00 00 00 00 00 00 00
0818- 00 00 00 00 00 00 00 00
0820- 00
```

Remembering that the nubers 10 through 15 are represented by the letters A through F in Hex, and that the 6502 represents addresses low byte first, high byte second, we can understand the program structure by interpreting the first two bytes of the program 07 08 (starting at address $801) as the address $0807 of the next program line, which we haven't yet written. We can also see that the linenumber is 10 by looking at the next two bytes at $802 - 803, (containing 0A 00, low-high) and that the only command token in the line is $80 representing the END statement.

We can change the token by typing:

```
*805:81 (RET)
```

if we get back to Applesoft by typing Control-C, and LIST our program we see:

```
]LIST

10  FOR
```

Which means that the Hex 81 we typed in at the monitor was understood by the interpreter as the FOR part of a FOR-NEXT statement. We can add a program line and see how things build in memory. For example:

```
]20 END
```

If we LIST the program:

```
10  FOR
20  END
```

Getting into the monitor (CALL -151) and listing memory $800.820 as before:

```
*800.820

0800- 00 07 08 0A 00 80 00 0D
0808- 08 14 00 80 00 00 00 00
0810- 00 00 00 00 00 00 00 00
0818- 00 00 00 00 00 00 00 00
0820- 00
```

We can derive the address of the second program line from $801 - 802, namely $807, and, reading from there, ($807) we see the address of the third program line (not yet written) at $807 - 808, namely $80D, which is the end of our program. The next two bytes (at $809 - 80A) are the hex representation of the line number and contain 14 00. We get the decimal linenumber by converting the two hex significant digits; multiplying the hex 1 by 16 and adding the least significant hex digit, in this case 4, to give (16 + 4) or 20 decimal as our linenumber.

We can contrast the token representations with other program contents by getting back to Applesoft with a Control-C and changing our program to:

```
]20 END : PRINT "HELLO"
```

If we type LIST:

```
10  FOR
20  END : PRINT "HELLO"
```

Getting into the monitor once again and looking at $800.820, we see:

```
0800- 00 07 08 0A 00 80 00 16
0808- 08 14 00 80 3A BA 22 48
0810- 45 4C 4C 4F 22 00 00 00
0818- 00 00 00 00 00 00 00 00
0820- 00
```

The address of the start of the third (unwritten) program line contained in bytes $807 - 808 has changed to $816, the hex representation of the linenumber at bytes $809 - 80A remains unchanged, as does the token for the END statement, $80. The colon is 3A or decimal (3*16 + 10) 58. getting the Apple to PRINT CHR$(58) should return a colon. Similarly the hex 22's (decimal 34) at the beginning and end of the ASCII representation of HELLO are the quotation marks required by the PRINT statement, while the PRINT command is represented by the token BA (hex).

If we can find some, we can get 'em all, and, taking some things as givens, (like the fact that Applesoft code occupies from $D000 to $F800 in ROM, and that the keyword dictionary starts at $D0D0) we can make Applesoft print its own keywords.

Before we do that, though, let's practice a few tricks. We know that 8 bit processors (such as the 6502) normally have 16 bit address busses accessing 64K of memory at any one time. We also know that a memory address consists of two bytes or four hex digits and that we need to convert those addresses to decimal if we want to use Applesoft to read itself. Converting four digit Hex addresses to decimal can be simplified if we remember this formula - the high digit is multiplied by 4096 and added to the next digit multiplied by 256 and added to the next digit multiplied by

16 added to the least digit multiplied by one.

To convert: $D0D0 = (13 * 4096) + (0 * 256) + (13 *16) + (0 * 1)

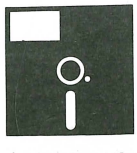
where the decimal equivalent of the hex digits can be summarized as:

```
01 02 03 04 05 06 07  08 09 0A 0B 0C 0D  0E 0F  hex
01 02 03 04 05 06 07  08 09 10 11 12 13  14 15  decimal
```

Getting back to Applesoft with Control-C, and clearing memory with NEW, we can list all of Applesoft's token dictionary and error messages by running the following program:

```
10  START = (13 * 4096) + (0 * 256) + (13 * 16) + (0 * 1)
20  FIN = START + 660
30   FOR I = START TO FIN
40  Y = PEEK (I)
50   PRINT CHR$ (Y);
60  NEXT I
```

Remember to include the semicolon after the PRINT statement in line 50 (to supress the carriage return after a PRINT statement), or else things will be impossible to follow. Next month we'll see how to customize the Applesoft command dictionary and create our own keywords.



FROM 5

If anyone is still trying to use the MASM assembler from the back of the August DOM, you may want to copy the second assembler disk which has the latest version of TED II we could find. TED II is easier to use, has a much better editor, and uses the same command syntax as Merlin.

As an effort to improve library access for those who cannot make it to the Core Meets, library copies to your disks can now be requested by mail order.

Please follow these guidelines:

1) Select the disk(s) you want from the catalog and/or the list in this issue of the Seed.

2) Label your blank disks with the disk selections you want (you may prefer to use both sides of the disk).

3) Insert the disks in a returnable mailer. The same mailer will be used to return your disks.

4) Enclose return postage, stamps only, for the same amount as the sending postage.

5) Enclose a self addressed return address label.

6) Calculate the copying fee to enclose at $1.00 per disk side copied (except DOM's within 12 months which are $4.00 to your disk).

7) Enclose payment in full by check or money order for all your copies.

8) Send to: Ed Thompson
Librarian, Denver Apple Pi
11768 East Atlantic Place
Aurora, Colorado 80014

9) Allow 4 weeks for turnaround. Disks will be handled as quickly as we can. You do not need to provide proof of membership, we will check the club membership list.

Ed Thompson, Librarian

AppleWorks

WHAT'S NEW--

Along with other annoucements from Apple Computer Inc. this month is an annoucement of the latest version of AppleWorks, version 2.0. The latest version will support the new expanded memory of the GS and will improve some of the problems that people have had with the current version. One of the big features is Mail Merge between the data base and word processor (yah!!). The spread sheet will support logical AND's and OR's. If you own AppleWorks now you will be able to upgrade it for $50.00 to the new version. The 2.0 will be available around the first of the year.

HANDS ACROSS THE CITY--

Another local Apple support group, C.A.C.H.E., is having members of the AppleWorks Product Management Group of Apple Computer Inc. as guest speakers at their October meeting. Rumor has it that Rupert Lissner (AppleWorks' author) may attend. C.A.C.H.E. will hold this meeting on Oct. 9th in the Denver University General Classroom Auditorium at 6:30pm. If you plan to attend you must RSVP with Betty Ketterling at 364-0752.

NATIONAL APPLEWORKS USERS GROUP--

If you have been looking for support for AppleWorks you may want to contact the National AppleWorks Users Group (NAUG). This group puts out a monthly newsletter, has a BBS and a library all for AppleWorks users. Dues are $24.00 per year and you can get a sample newsletter for $3.00. For more information contact:

NAUG
BOX 87453
Canton, MI 48187

BBS PHONE NUMBER

422 - 3716

HOT LINE

422 - 0005

ORDER FORM
Sept, 1986

Denver Apple Pi
P.O. Box 17467
Denver, Colorado 80217

ship to:_____

_____

Category	Sub-Category	Circle disk numbers desired from list below.
Bus. & Database	Business	111 112 113 114 115 116 117 118 119 120 131 151 152 153
	Business Mgmt	132 133
	Database	201 202 203 204 205 206
	Engineering	301 302 311 331
	Science	351 352 353 354 355 356 357 371 372 373 374 375 376 377
Language & Tutor	Computer Lang	401 402 403 404 405 406 407 408 409 410 411
	Computer Tutor	451 452 453 454 455 456 461 462 463 464 465 466 467 468
Music	Music	471 472 473 474 475 476 477 478
Education	Foreign Lang	501 502 503 504
	Math Programs	511 512 513 514 515 516 517 518 519 525 526 527 551 552 553 554 555
	Reading & Lang	541 542 543 544
	Science	561 562 563 564 565 571 572
	Education	581 582 583 584 585 586 587 588 589 590 591 592 593 594
Personal & Home	Home	601 602 621 622 631 632 641 642 643 644 645 646 647
Utilities	Utilities	675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690
Eamon Games	Eamon	700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 763 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799
Games	Games	801 802 803 811 812 813 814 815 816 821 822 823 827 828 829 830 831 832 837 838 839 840 841 847 848 849 850 851 852 853 861 862 863 864 870 871 872 873 874 875 876 877 878 879 880 881 891 892 893 894 895 896 897 898
X-Rated	X-Rated	891 892 893 894
Communication	Hobby/Radio	901 902 903 911 912 913 914 915
Graphics & Shapes	Graphics	931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976
Inter. Apple Core	IAC	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 45 46 47 48 53 54 56 57 58 41 45 46 47 48 53 54 56 57 58
Disk of the Month	DOM	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
Personal Domain	Personal Domain	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Appleworks	Appleworks	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 60 61
Prodos	Prodos	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Pascal	Pascal	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
CP/M	CP/M	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198

There is a 10 copy side/member/day limit. Disks-of-the-month for the most recent 12 months are purchased for $5.00 each and are not copied. All other copies are $1.00/side or $2.00/side with disks.

Total DOM'S _____ x $5.00 = _____

Total Copies _____ x $1.00 = _____

Total Disks _____ x $1.00 = _____

Total Cost ..... _____

Note: Copies can be made to both sides of a disk. Also, there is no copy charge when a program is submitted with each copy request!

REMEMBER THE GARAGE SALE AT THE CORE MEET

SUNDAY OCT. 5TH

BRING:

HARDWARE

SOFTWARE

COMPUTER GOODIES

OR

$$$

THE SEED

denver apple pi

P.O. Box 17467
Denver, CO 80217

First Class
U.S. POSTAGE
PAID
DENVER, CO
PERMIT- 3182